

APPENDIX

CODING INITIAL DECLARATION

```
1 #include <ESP8266WiFi.h>
2 #include <WiFiClientSecure.h>
3 #include <UniversalTelegramBot.h>
4 #include <math.h>
5 #define R 6371
6 #define TO_RAD (3.1415926536 / 180)
7 const char* ssid = "agusetiawan";
8 const char* password = "123456789";
9 #define BOTtoken "1313160602:AAEYzOv-kFHTP2EVYXUI5bXTwrqgUPG18S4"
10 #define CHAT_ID "1398090082"
11 WiFiClientSecure client;
12 UniversalTelegramBot bot(BOTtoken, client);
13 const int motionSensor = 14; // PIR Motion Sensor
14 bool motionDetected = false;
15 void ICACHE_RAM_ATTR detectsMovement() {
16 motionDetected = true;
17 }
18 struct LatLong {
19 double latitude;
20 double longitude;
21 };
22 // Jarak LatLong RUMAH --> POS 1 110
23 LatLong Rumah = {-7.0529222, 110.4226965};
24 LatLong POS1 = {-7.052325, 110.422260};
25 // Jarak LatLong RUMAH --> POS 2 120
26 LatLong Rumah1 = {-7.0529222, 110.4226965};
27 LatLong POS2 = {-7.052219, 110.422349};
28 // Jarak LatLong RUMAH --> POS 3 130
29 LatLong Rumah2 = {-7.0529222, 110.4226965};
30 LatLong POS3 = {-7.052210, 110.422368};
31 // Jarak LatLong RUMAH --> POS 4 100
32 LatLong Rumah3 = {-7.0529222, 110.4226965};
33 LatLong POS4 = {-7.052372, 110.422226};
34 // Jarak LatLong RUMAH ---> POS 5 100
35 LatLong Rumah4 = {-7.0529222, 110.4226965};
36 LatLong POS5 = {-7.052382, 110.422103};
```

CODING LOOP

```
37 {
38 jarak[0] = distance(Rumah.latitude, Rumah.longitude, POS1.latitude,
39 POS1.longitude);
40 jarak[1] = distance(Rumah1.latitude, Rumah1.longitude, POS2.latitude,
41 POS2.longitude);
42 jarak[2] = distance(Rumah2.latitude, Rumah2.longitude, POS3.latitude,
43 POS3.longitude);
44 jarak[3] = distance(Rumah3.latitude, Rumah3.longitude, POS4.latitude,
45 POS4.longitude);
46 jarak[4] = distance(Rumah4.latitude, Rumah4.longitude, POS5.latitude,
47 POS5.longitude);
```

```

43 Serial.print("Distance in RUMAH TO POS 1: ");
44 Serial.print(jarak[0], 3);
45 Serial.println("m");
46 Serial.print("Distance in RUMAH TO POS 2: ");
47 Serial.print(jarak[1], 3);
48 Serial.println("m");
49 Serial.print("Distance in RUMAH TO POS 3: ");
50 Serial.print(jarak[2], 3);
51 Serial.println("m");
52 Serial.print("Distance in RUMAH TO POS 4: ");
53 Serial.print(jarak[3], 3);
54 Serial.println("m");
55 Serial.print("Distance in RUMAH TO POS 5: ");
56 Serial.print(jarak[4], 3);
57 Serial.println("m");
58 double min=10000000000;
59 for (int a=0;a<jumlah_posisi;a++){
60 if (jarak[a]<min){
61 min=jarak[a];
62 }
63 }
64 Serial.print("jarak minimal: ");
65 Serial.print(min, 3);
66 Serial.println("m");
67 motionDetected= digitalRead(motionSensor);
68 Serial.println("Data Pir:"+String(motionDetected));
69 if (motionDetected){
70 String protokol = "\nMotion Detected\nJl Mawar No. 10\nJarak Minimum:
    "+String(min)+" M";
71 bot.sendMessage(CHAT_ID, protokol, "");
72 Serial.println("Motion Detected");
73 motionDetected = false;
74 }
75 }

```



0.12% PLAGIARISM
APPROXIMATELY

Report #13361627

Introduction Background In this project I want to solve a problem that often occurs, namely robberies when the house is empty. Because currently, his house is unoccupied, it is often the target of theft. The current theft cases do not take into account time and place. Because nowadays there are often cases of theft that occur during the day and the environment is quiet. The case of theft can occur because the security system in the home environment is still lacking, such as the number of security officers who are still small, the condition of the house environment is quiet so that they are able to commit criminal acts easily without someone knowing. In solving the problem of this project, I used the Haversine algorithm approach to find the nearest location in sending data in the form of notifications received via smartphones. So with the haversine algorithm and the data structure that I use, I can draw final conclusions in generating comparisons to determine the effect of the results on the data transmission process. So